Searchable Patent Embeddings

Lawrence Liu
lwl@mit.edu

Aaron Guo
aaguo@mit.edu

Joyce Qu
joycequ@mit.edu

Abstract

Patents are critical to inventors, businesses, and governments. They protect valuable intellectual property by creating guardrails aside similar innovations. Patent landscaping is the process of developing a comprehensive set of patents related to a specific topic by iteratively finding similar patents. We propose 3 unique patent embedding strategies and 2 novel evaluation metrics to build a system to match patents. Over a sample of 30 thousand patents published in 2023, we show that the summarized fulltext description section of each patent, which has been underutilized by prior work, produces the most informative patent embeddings.

1 Introduction

As the total number of patents granted each year continues to increase (USPTO, 2020), there is a greater need to more accurately and efficiently search the universe of patents. Understanding and protecting intellectual property (IP) is crucial to a large part of the US economy. A 2019 report by the US Patent and Trademark Office (USPTO, 2019) reported that 41% of U.S. economic activity, representing over 63 million jobs, occur in IP intensive industries. The economy around issuing, enforcing, and dealing in patents is also worth tens of billions of dollars. Patent data provides pinpoint indicators of a company's material activities and innovative output, numbers which remain hidden in reported R & D expenditures (Lerner and Seru, 2022). It is clear that there is a need to efficiently and accurately check for similar ideas (patent landscaping) to protect IP.

Patent landscaping is the process of mapping out all relevant patents for a particular topic. There is no widely accepted best method for landscaping since use cases differ. Oftentimes, patent landscaping is a bespoke, manual process which aims to build a network of connected patents through complex queries on patent databases. However, there are two obvious problems with this approach. First, there is a non-trivial level of field-specific expertise needed to determine if any two patents are related. Expert evaluation is expensive and inaccessible in niche fields. Second, there are currently millions of outstanding patents and hundreds of thousands of new patents granted each year (USPTO, 2020). The current USPTO advanced search function is ineffective (a search for "garden AND hose" yields top results including a scent diffusion device and a shower head). Developing techniques that can search more of the patent universe efficiently can greatly improve the effectiveness of patent listing inspections.

There have been many efforts to develop automated patent landscaping processes using deep learning (Abood and Feltenberger, 2018; Choi et al., 2019). These models have been demonstrated to be effective ($F_1 \geq 0.9$) for the task of developing patent landscapes around broad topic prompts (i.e. "operating system", "browser", and "machine learning"), but they require starting with a seed set of hundreds, if not thousands of patents. Developing this seed set can be just as difficult, if not more so, than expanding it. (Erana and Finlayson, 2024) build on these works by developing a more detailed model that takes into account the entire patent text. This allows them to decrease the initial size of seed set down to as low as 24 patents while still maintaining similar accuracy.

We propose a new strategy of searching patents that allows users to start building a patent landscape with a single patent. By embedding patents in a semantically informed way, each patent can be compared with all other patents to measure similarity. This may create a more context aware matching scheme than current strategies. We also evaluate the relative effectiveness of using various elements of patents in training embeddings, informing future research of the trade off between computation costs and model potency.

Using this strategy of embedding and creating patent landscapes, we develop a simple tool that can help a user visually grow a patent landscape from an initial patent by providing the top K most similar patents to any patent. Using this resource, users may quickly develop a network of similar patents to find one patent that is close to one of the search results. Compared with the current USPTO search, this is more efficient and more comprehensive, expediting and improving the breadth of the search.

2 Related Work

Both patent landscaping and patent classification tasks are active fields of research.

Much of patent analysis from the early 2000s to 2019 have focused on patent classification tasks. Standard developments in patent classification include DeepPatent and PatentBERT. DeepPatent (Li et al., 2018) is a deep learning algorithm based on CNN for images in patent information and word vector embedding, followed by evaluation with CLEF-IP, a standard patent classification dataset. PatentBERT (Lee and Hsiang, 2019) finetunes and applies a pre-trained BERT model to patent classification and requires only the patent claim for classification.

A contemporary encoding model with sequentially learned context-based embedding layers that measures up to the density of a patent corpus can be found in Contextual Document Embeddings by Morris and Rush (2025). The novel two-pass encoder offers a readily computable, generalist baseline to compare to patent-fine-tuned encoders such as PatentBERT.

Although there has been significant progress with patent classification tasks, the breadth of patent landscaping is rather sparse in comparison. A typical heuristic for patent landscaping is to test a machine learning classifier on an assembly of custom patent data keywords as classes (Erana and Finlayson, 2024). Benchmark datasets are often explicitly defined for these tasks, such as narrow datasets to the likeness of "Marine Plant Using Augmented Reality Technology" in Choi et al. (2019) and broader categories like "drugs and medical" and "computers and communication" in the National Bureau of Economic Research (NBER) patent citation dataset (Lerner and Seru, 2022).

Patent similarity and similar patents retrieval lays the foundation for patent landscaping and patent analysis. One of the earlier patent matching papers uses tf-idf embeddings for all patents by scraping the complete information for each patent from USPTO site and calculating n^2 similarity scores (Younge and Kuhn, 2016). This brute-force approach is computationally expensive but it has been cited by later papers as a relatively effective technique. In order to use a smaller sized representation of each pattern and decrease the computational cost, previous works have pointed to extracting important bits of the patent rather than its long-form to train models for patent matching (Sinha et al., 2021). Due to simplifying truncations (Erana and Finlayson, 2024), there has been little utilization of the bulk of patent data found in the full text (detailed description) of a patent.

Patent landscaping enables potential patent authors to discover efficiently and exhaustively related patents and patent applications to their product. This is generally referred to as prior-art search. Creating embeddings has thus focused on the purpose of what is called patent matching.

Using semi-supervised machine learning, from a seed set of patents, the patents can be expanded by considering patents in the same patent family and the same subgroup code to produce a larger starting set (Abood and Feltenberger, 2018). This makes it easier to search the patent universe since the starting set is considerably broader. The customization of benchmark seed sets and obtaining expert labels is contemporary precedent for patent landscaping.

More experimentation was applied in model architecture, going beyond multi-layer perceptrons as used in (Abood and Feltenberger, 2018) to multi-head self attention and convolutional neural networks (Choi et al., 2019). These models showed improvements over prior results. Most recently, models have been focused on fine-tuning, incorporating more data from patents, and incorporating Cooperative Patent Classification (CPC) labels (Pujari et al., 2022). Extending embeddings beyond title and abstract corpuses is explicitly recommended.

We aim to extend the existing knowledge on patent matching by experimenting with summarization to improve landscaping results in efficiently exploring a space of similar patents to a single patent.

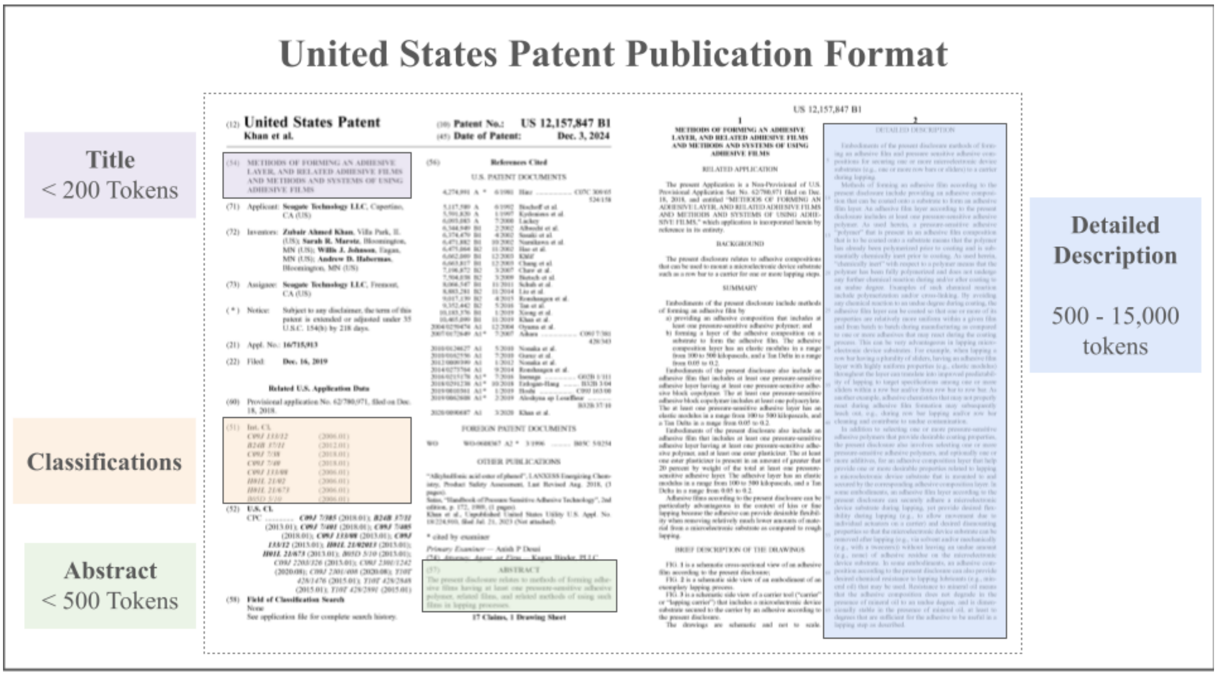

Figure 1: Layout of United State Patent Publication. The title and abstracts are typically very short relative to the detailed description section.

3 Methods

The paper's objectives pertain to the creation of a float vector embedding for each patent, enabling classification accuracy tests and pairwise similarity comparisons over the entire patent corpus.

3.1 Data

To obtain data for this research, we downloaded data directly from the US Patent and Trademark Office (USPTO). The USPTO Bulk Data Storage System (BDSS) is a repository of all US patent applications and publications back dating back to January of 1971. The repository is updated weekly with the most recent patent publications.

For this project, due to constraints with compute and time, we restricted the patent universe to patents published in 2023, of which there were 616 thousand. These were stored in the "Patent Application Full Text Data (No Images)" section on the BDSS. From the patent application files, we separated individual patents and extracted the four main sections depicted in the Figure 1 alongside a unique ID for each patent. All other sections of the patent were either additional labels or abridged versions of detailed description text. We chose to include the abstract instead of the summary as an abridged description of the patent since prior works (Abood and Feltenberger, 2018; Choi et al., 2019; Erana and Finlayson, 2024) utilized abstracts as the main description.

We sample approximately 30 thousand patents from the 616 thousand patents published in 2023 to fall within the bounds of compute cost. This provides a dense set for each class, as the least occurring section letter, 'D' (Textiles and Paper), has 140 instances in the sample.

Parameter	Value
Number of patents	616840
Number of patents sampled for study	29536
CPC section (classes)	9 + 1 (no section)

Table 1: Detailed Dataset Lengths

3.2 Architecture

We extracted three types of text from each patent: title, abstract, and full text description. There is a large mismatch between the text length of each. Table 2 displays the token lengths calculated using the patentBERT (Lee and Hsiang, 2019) tokenizer. Each text section is at least an order of magnitude larger or smaller than another text section. Therefore, we treat each of these sections differently.

Since the title is, on average, only about 11 tokens, we concatenate the abstract text to it. We refer to this concatenated text as "titleabstract."

To embed the titleabstract, we utilized patentBERT, a BERT model fine tuned for patent classi-

Text	Min	Max	Average
Title	3	101	11.4
Abstract	7	607	137.4
Fulltext	599	983606	15341.9

Table 2: Min, Max, and Average token length of patent text sections using the patentBERT tokenizer

fications (Lee and Hsiang, 2019). Akin to BERT, patentBERT's maximum input length is 512 tokens. Since the titleabstract unlikely to exceed 512 tokens, it could be embedded as is. In the cases when the titleabstract was too large, we truncated to the first 512 tokens. This did not cut out more than 200 tokens. From the patentBERT model, we extracted the final layer, leaving us with a (1024,) vector embedding of titleabstract for each patent.

To establish a baseline, we also embedded titleabstract using Contextual Document Embeddings (CDE)(Morris and Rush, 2025). CDE is a generalized document encoder that enables explicit encoding of neighboring document information. The encoder takes a maximum input length of 512 tokens. The final layer is a (768,) vector embedding of titleabstract for each patent.

The model architecture for CDE was transmitted from an applied implementation of the model (where the neighboring document information is a sample of the corpus itself): CDE - fiqa demonstration. The model thus gets two chances to learn the document information, one from the context embedding for a smaller random sample of the *titleabstract* corpus, and another from embedding the full corpus with context. Due to cost constraints, only titleabstract was prepared on the CDE model.

We take more precaution in dealing with the length of the fulltext. To embed the fulltext, we utilized two strategies denoted "truncated" and "summary," both of which are encoded by PatentBERT (see Figure 2). The first strategy, "truncated," simply truncates the fulltext and only pays attention to the first 512 tokens. The truncated fulltext is then embedded in the same way as the titleabstracts. This is similar to how (Erana and Finlayson, 2024) processed the fulltext. However, when the average fulltext is over 15,000 tokens, selecting just the first 512 tokens misses the majority of the information contained in fulltext. The second strategy, "summary," attempts to embed more of the fulltext by summarizing the fulltext. Using the distilBART implementation of BART (Lewis et al., 2020), we summarize each 1024 token section of the text into sections of at most 100 tokens. We continuously append summarized sections until we obtain 512 total tokens of summary. The summary text is then embedded in the same was as titleabstracts. The embedding is minimally processed afterwards using layernorms to stabilize the vectors. In the end, we end up with 4 embedding schemes:

1. *Titleabstract*: Embedding of shape (1024,), created from exclusively the titleabstract embedding

2. *Fulltext Truncated*: Embedding of shape (2048,) created from concatenating the title abstract and truncated fulltext embeddings

3. *Fulltext*: Embedding of shape (2048,) created from concatenating the titleabstract and summarized fulltext embeddings

4. *Titleabstract (CDE)*: Embedding of shape (768,) created from exclusively titleabstract using intermediate embedded context to generate embeddings

Embedding strategies 2 and 3 are summarized in Figure 2.

3.3 Evaluation

There is precedence in measuring the quality of patent classification. Previous models have used binary classification F_1 scores on a key term 'AI' in Abood and Feltenberger (2018) or four technology areas based on key phrases like 'reverse conductive' and 'mini dipole' in Choi et al. (2019).

We leverage the existing patent classification schemes published by the USPTO as class labels. The Cooperative Patent Classification (CPC) scheme is a multilevel classification scheme. At the highest level, classes are labeled by section letters A through H as seen in Table 3. Patents can take on multiple labels from various categories, meaning very unique patents can belong in the same category. Patent classification categories by USPTO vary based on need to include new technologies and to achieve better resolution among categories, so we trust that patents in the same category are more similar to other patents that have at least one classification code with the same section letter than patents that do not share any labels (Toole et al., 2020).

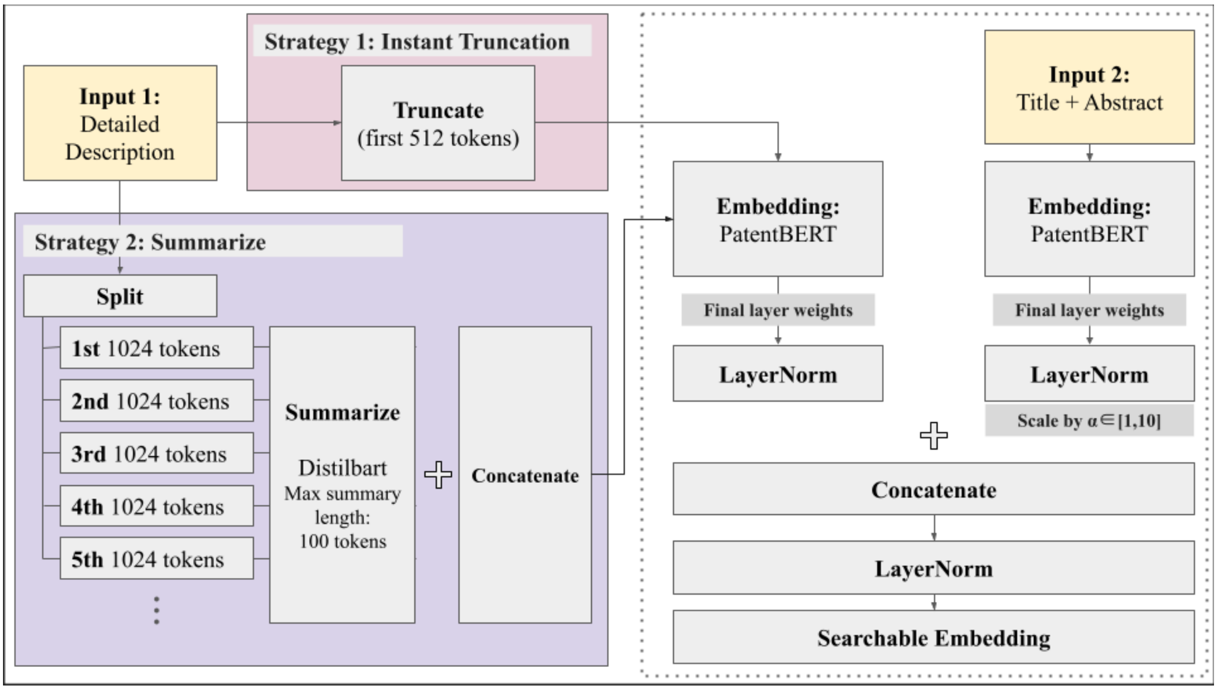

Figure 2: *Fulltext Truncated* and *Fulltext* Model Architecture

Code	CPC Section (Category)
A	Human Necessities
B	Performing Operations; Transporting
C	Chemistry, Metallurgy
D	Textiles, Paper
E	Fixed Constructions
F	Mechanical Engineering; Lighting; Heating; Weapons; Blasting
G	Physics
H	Electricity
Y	General Tagging of New Technological Developments

Table 3: CPC classification codes: Section

We use the F_1 evaluation metric to evaluate performance on the patent classification task, in concurrence with previous work. For each of the nine lettered CPC sections (plus the no label section) and for each of the embeddings, we train a SVM (Support Vector Machine) classifier. Since each patent in the sampled corpus may belong to several CPC codes and thus several classification classes, the sample average of F1 takes the 10 times applied binary classification F1 metric into account for this per-patent multiplicity.

There is no objective test set that specifies which measures the quality of a patent landscape. We develop our own evaluation metrics to analyze the patent landscaping task. We aim to quantify how much more similar patents sharing a CPC classification section are compared to patents that do not share any CPC classifications.

We base our evaluation metrics off this heuristic and develop two evaluation metrics to compare embedding strategies - *closeness* and *same category similarity*.

To establish general notation, let the set of patents analyzed be P, with $|P| = n$. The cosine similarity between each patent p_i and all other patents can be expressed as a $n \times 1$ vector denoted s_i. Sorting the s_i in descending order yields the most similar patent in index 0 and the least similar patent in index $n-1$. Let this sorted s_i be denoted s_i'. For each $p \in P$, $s_i'[p]$ returns the index of p in s_i'; larger indices indicate lower similarity.

Let $c(p)$ be the set categories that p is labeled with. All patents belong to at least one category. The set of all patents p' that share at least one category with p can be denoted as

$$P(\mathrm{c}(p)) = \{p' \in P \mid \mathrm{c}(p) \cap \mathrm{c}(p') \neq \emptyset\}$$

Using these definitions, we formalize the *closeness* and *same category similarity* evaluation metrics.

Closeness: A measure of the average proportion of same category patents that are in the top half of similarity.

$$A(p_i) = \frac{1}{|P(c(p_i))|} \sum_{p' \in P(c(p_i))} \mathbb{1}\left(s_i'[p'] < \frac{n}{2}\right)$$

$$Closeness = \frac{1}{n} \sum_{i=1}^{n} A(p_i)$$

Same category similarity (SCS): A measure of the proportion of patents that have a majority of same category category patents in the top half of similarity rankings.

$$SCS = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(A(p_i) > \frac{1}{2}\right)$$

A larger $closeness$ metric and larger SCS metric across all patents for a particular embedding indicate that the embedding is better at recognizing similar patents, in terms of CPC Section codes.

We will use F1 score for classification, and we will use closeness and SCS landscaping to compare over all 4 embedding schemes described in section 3.2.

4 Results

All following analyses are based off the corpus of 30 thousand patents.

Using the prior defined metrics of $closeness$ and SCS, we evaluate the efficacy of *Titleabstract*, *Fulltext Truncated*, and *Fulltext*.

Embedding	Closeness
Titleabstract	0.599
Fulltext Trunc.	0.592
Fulltext	0.604
Titleabstract (CDE)	0.632

Table 4: Closeness metrics on embedding schemes

Embedding	SCS
Titleabstract	0.841
Fulltext Trunc.	0.848
Fulltext	0.893
Titleabstract (CDE)	0.904

Table 5: SCS metrics on embedding schemes

The (summarized) fulltext embeddings proved marginally more accurate than its patentBERT-based counterparts. The fulltext embedding was the most effective among the patentBERT encodings, outperforming in the closeness and SCS metrics, significantly outperforming the two others in the SCS metric. We visualize the distribution of the individual $A(p)$ that form the SCS in Figure 3. The distribution of the fulltext embeddings is more skewed left than the other embeddings, with a larger portion of same category patents are in the top 50% of patents (right of the red dashed line) when ranked by cosine similarity.

Between the two Titleabstract embeddings, the generalist CDE encoder outperforms the task-specific patentBERT by substantial margins in clossness and SCS. The CDE encoder also marginally outperforms Fulltext summarized encodings using patentBERT.

Under classification tasks, all patentBERT encoders significantly outperforms its generalist contemporary, the CDE encoder in the classification F1 metric.

Embedding	F1-score Sample Avg
Titleabstract	0.78
Fulltext Truncated	0.78
Fulltext	0.79
Titleabstract (CDE)	0.67

Table 6: Sample average F1 scores for classification on embedding schemes

5 Discussion

From the results, we see that the fulltext embedding performed the best across both evaluation metrics relative to the other two embeddings based on Patent-BERT-based encodings. There are multiple ways to interpret this result.

From the evaluation metric side, the evaluation metrics use 50% of the sample as a cutoff for similarity for a variety of purposes. Firstly, the CPC codes are non perfect; each code is updated periodically as technologies change and shift. These updates are not done across all categories at the same time. Secondly, there is significant diversity within each category; there are inevitably times when out of category patents are more similar than the least similar in-category patents. the 50% cutoff was chosen because it represented the heuristic that same category patents should be more similar than different. Thus, a higher $closeness$ and SCS indicate that the embedding better represents

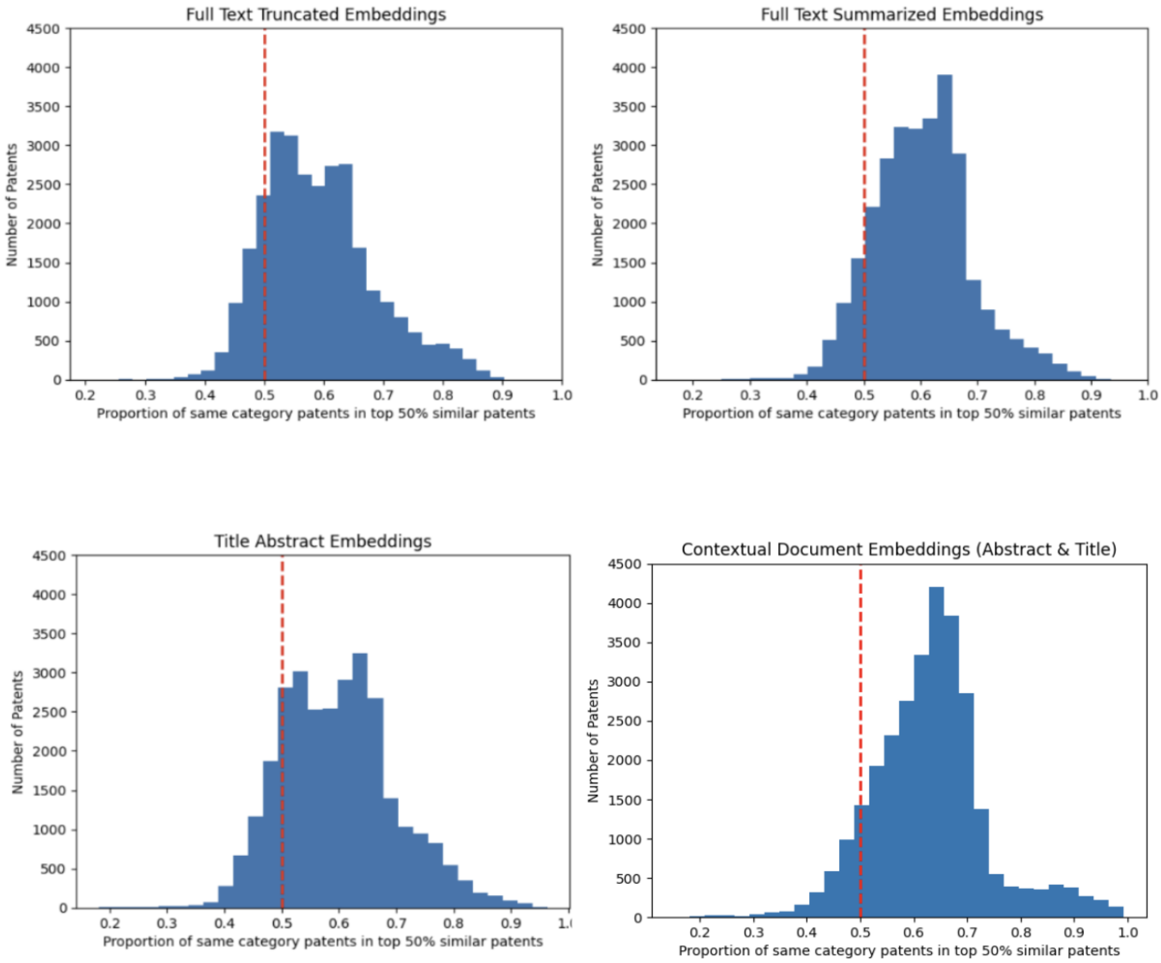

Figure 3: Distribution of $A(p)$ used in the SCS calculation

the underlying connections between patents, even when the embedding was not specifically informed of patent classifications.

From the relative performances, the results also indicate that relevant information is not concentrated in the beginning of each fulltext. The performance of Fulltext Truncated and the Titleabstract were very similar in both $closeness$ and SCS. It is necessary to consider a longer portion of the fulltext to realize improvements to the embedding understanding. However, this does come at a cost. Summarizing approximately 5000 tokens of a 15,000 token fulltext took approximately 10 times as long as simply truncating the first 512 tokens on the same architecture. An intuitive explanation for why truncation is occasionally less effective than titleabstract embeddings comes from the type of content at the beginning of the fulltext. Looking through a few dozen fulltext descriptions by hand, we notice that fulltext descriptions typically start with a section that just lists metadata, ie who the inventors were, their locations, related patent numbers etc. This contains very little information about the content of the patent itself and contributes little to the quality of the embedding created.

From the discrepancy in classification F1 scores between the patentBERT based models and the CDE model, we can see that using a fine tuned model for embeddings may have contributed to improved performance. Intuitively, this may be because patents are highly technical documents that have a significantly larger vocabulary than general document embeddings may cover.

While most recent works in patent landscaping do not use the fulltext, or simply truncate the fulltext to the first 512 tokens, our results indicate that there is valuable information and context contained in the fulltext descriptions. Future research could explore more diverse strategies of summarizing the fulltext or models that can embed very long inputs. On possible strategy to explore is that instead of summarizing from the beginning of the fulltext un-

Classification Results (SVM)

Figure 4: Fulltext Classification

section	precision	recall	f1-score	support
A	0.83	0.77	0.80	701
B	0.71	0.64	0.67	848
C	0.80	0.82	0.81	371
D	0.74	0.50	0.60	28
E	0.77	0.65	0.71	229
F	0.82	0.74	0.78	605
G	0.82	0.86	0.84	2564
H	0.85	0.81	0.83	2249
Y	0.00	0.00	0.00	20
None	0.85	0.03	0.05	667
Samples Avg	0.83	0.80	0.79	8282

Figure 5: Fulltext Truncated Classification

section	precision	recall	f1-score	support
A	0.82	0.77	0.79	701
B	0.73	0.62	0.67	848
C	0.79	0.79	0.79	371
D	0.72	0.64	0.68	28
E	0.77	0.69	0.72	229
F	0.85	0.73	0.79	605
G	0.83	0.84	0.84	2564
H	0.84	0.80	0.82	2249
Y	0.10	0.05	0.07	20
None	0.94	0.02	0.05	667
Samples Avg	0.83	0.79	0.78	8282

Figure 6: Titleabstract Classification

section	precision	recall	f1-score	support
A	0.81	0.76	0.78	701
B	0.69	0.63	0.66	848
C	0.79	0.83	0.81	371
D	0.79	0.54	0.64	28
E	0.74	0.66	0.70	229
F	0.83	0.73	0.77	605
G	0.84	0.86	0.85	2564
H	0.85	0.80	0.83	2249
Y	0.03	0.05	0.04	20
None	0.85	0.03	0.05	667
Samples Avg	0.83	0.79	0.78	8282

Figure 7: Titleabstract (CDE) Classification

section	precision	recall	f1-score	support
A	0.77	0.56	0.65	689
B	0.72	0.42	0.53	896
C	0.82	0.76	0.79	403
D	0.57	0.15	0.24	26
E	0.81	0.34	0.48	220
F	0.79	0.58	0.67	606
G	0.80	0.80	0.80	2593
H	0.77	0.71	0.74	2127
Y	0.00	0.00	0.00	20
None	0.00	0.00	0.00	648
Samples Avg	0.72	0.68	0.67	8228

til the token limit is reached, one could consider summarizing all 1024 token segments and ranking their informativeness. This should provide an even more informative embedding.

6 Ethics Statement

While the proposed strategy of patent embeddings aims to capture similarities between patents, they should not be directly utilized for patent claims. The patent embeddings are purely suggesting the most semantically similar patents, which can have varying degrees of relationship to the patent of interest to the user. The patent embeddings may also not be a comprehensive map of all relevant patents because they consider patents as a whole; users may be interested in patents similar to a part of a specific patent, which has not been evaluated in this research. This work is most useful in developing a high level mapping of patents which, as a whole, are similar to some given patent. Our results primarily show that the fulltext descriptions in patents are more informative than title and abstracts. Future research in the field of machine learning for patents should focus more attention on the fulltext descriptions, despite its bulk.

7 Code

All code can be found on the github here: https://github.com/Bookmaster9/searchablepatents. Data files are too large to be uploaded directly to github, but can be extracted from the USPTO Bulk Data Storage System using files from patent_cleaning_extraction and the details in Section 3.1.

References

Aaron Abood and Dave Feltenberger. 2018. Automated patent landscaping. *Artificial Intelligence and Law*.

Seokkyu Choi, Hyeonju Lee, Eunjeong Park, and Sungchul Choi. 2019. Deep patent landscaping model using the transformer and graph embedding". *arXiv*.

Tisa Islam Erana and Mark A. Finlayson. 2024. Automated neural patent landscaping in the small data regime. *arXiv*.

Jieh-Sheng Lee and Jieh Hsiang. 2019. Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv*.

Josh Lerner and Amit Seru. 2022. The use and misuse of patent data: Issues for finance and beyond. *The Review of Financial Studies*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Shaobo Li, Jie Hu, Yuxin Cui, and Jianjun Hu. 2018. Deeppatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117.

John X. Morris and Alexander M. Rush. 2025. Contextual document embeddings. pages 1–14.

Subhash Pujari, Jannik Strötgen, Mark Giereth, Michael Gertz, and Annemarie Friedrich. 2022. Three real-world datasets and neural computational models for classification tasks in patent landscaping. *ACL Anthology*.

Priyanshu Sinha, Rishabh Tripathi, Sthita Pragyan Pujari, Rakesh Ch Balabantaray, Prabhjit Thind, and Satya Narayan Kar. 2021. Conceptual search for patent similarity match. In *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 1–5. IEEE.

Andrew A. Toole, Nicholas A. Pairolero, James Q. Forman, and Alexander V. Giczy. 2020. Patentbert: Patent classification with fine-tuning a pre-trained bert model. *Santa Clara High Technology Law Journal*.

USPTO. 2019. 2019 uspto performance and accountability report.

USPTO. 2020. U.s. patent statistics chart.

Kenneth A. Younge and Jeffrey M. Kuhn. 2016. Patent-to-patent similarity: A vector space model. *SSRN*.